# Hash Functions Bridging the Gap from Theory to Practice

**Mikkel Thorup**
University of Copenhagen, Denmark

## Abstract

Randomized algorithms are often enjoyed for their simplicity, but the hash functions employed to yield the desired probabilistic guarantees are often too complicated to be practical. Hash functions are used everywhere in computing, e.g., hash tables, sketching, dimensionality reduction, sampling, and estimation. Many of these applications are relevant to Machine Learning, where we are often interested in similarity between high dimensional objects. Reducing the dimensionality is key to efficient processing. Abstractly, we like to think of hashing as fully-random hashing, assigning independent hash values to every possible key, but essentially this requires us to store the hash values for all keys, which is unrealistic for most key universes, e.g., 64-bit keys. In practice we have to settle for implementable hash functions, and often practitioners settle for implementations that are too simple in that the algorithms end up working only for sufficiently random input. However, the real world is full of structured/non-random input. The issue is severe, for simplistic hash functions will often work very well in tests with random input. Moreover, the issue is often that error events that should never happen in practice, happen with way too high probability. This does not show in a few tests, but will show up over time when you put the system into production. Over the last decade there has been major developments in simple to implement tabulation based hash functions offering strong theoretical guarantees, so as to support fundamental properties such as Chernoff bounds, Sparse Johnson-Lindenstrauss transforms, and fully-random hashing on a given set w.h.p. etc. I will discuss some of the principles of these developments and offer insights on how far we can bridge from theory (assuming fully-random hash functions) to practice (needing something that can actually implemented efficiently).

**2012 ACM Subject Classification** Theory of computation

**Keywords and phrases** Hash functions

**Digital Object Identifier** 10.4230/LIPIcs.ISAAC.2025.2

**Category** Invited Talk

**Funding** *Mikkel Thorup*: Supported by VILLUM Foundation Grant 54451, Basic Algorithms Research Copenhagen (BARC).